

Braiding wires into the Linux network stack - recent work on embedded networking

Oleksij Rempel – ore@pengutronix.de

my_self = kzalloc()

- Oleksij Rempel, Linux Kernel Hacker
- Expertise in: Medical, Industrial and Agricultural devices
- Addressing challenges: Limited CPU/bandwidth, power efficiency, diagnostic
- Prioritizing long-term sustainable, secure and Open Source Embedded Linux (mainline).



Cool tech for a better future

A list of technologies to reduce resource consumption:

- To reduce the weight or amount of copper needed for the HW, use SPE (Single Pair Ethernet) if possible – 1000BaseT1, 100BaseT1, 10BaseT1L, 10BaseT1S.
- To reduce even more cables for power delivery, use PoDL (Power over Data Line) or PoE (Power over Ethernet)
- To reduce at least some of power consumption (~0.2W per port on my HW) use EEE



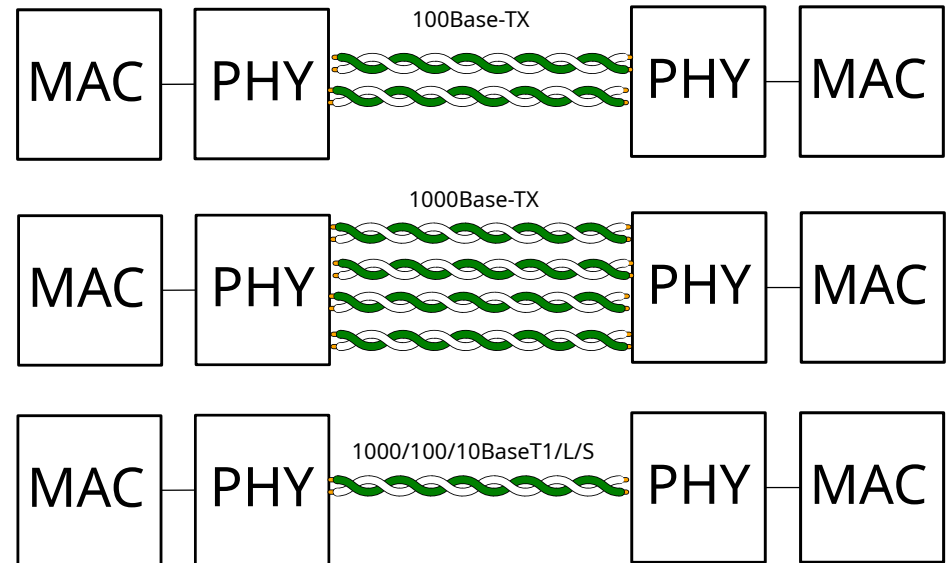
Calling All Hackers & Makers

- Hacker and Maker projects may benefit from this technologies:
- SPE and PoDL for home automation, 3D printing, robotics...



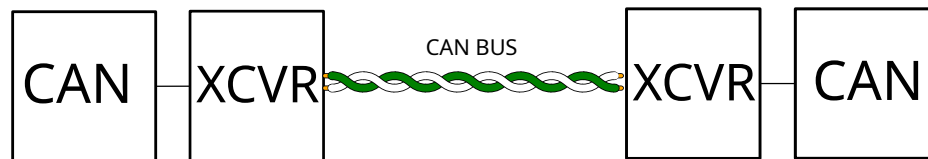
What is SPE?

- SPE is a collection of different standards: 10/100/1000/..../BaseT1/L/S
- Different optimizations for weight, length, link creation time, explosion Protection
- Ethernet but with one twisted pair

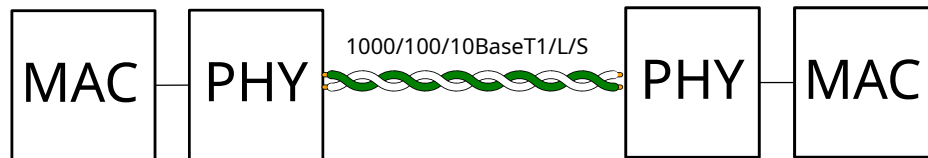


SPE vs ???

- New PHYs are needed
- Any? MAC can be reused.
- Reuse existing and adopted infrastructure and software
- Get QoS, IEEE 1588, bit migrating to SPE

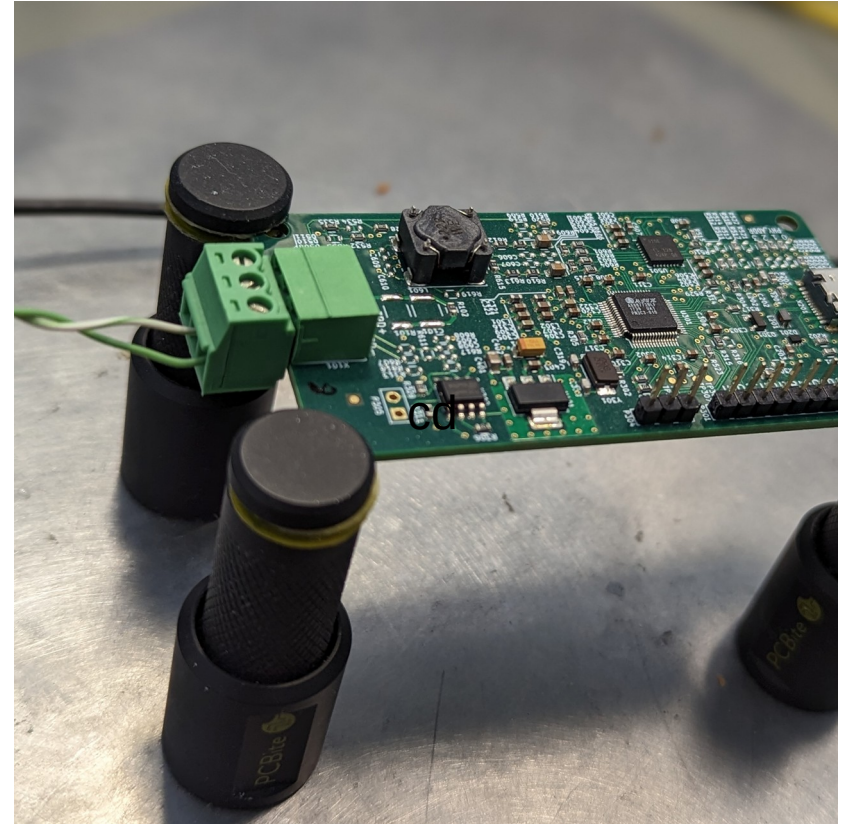


Most existing MACs can be combined with T1 PHY



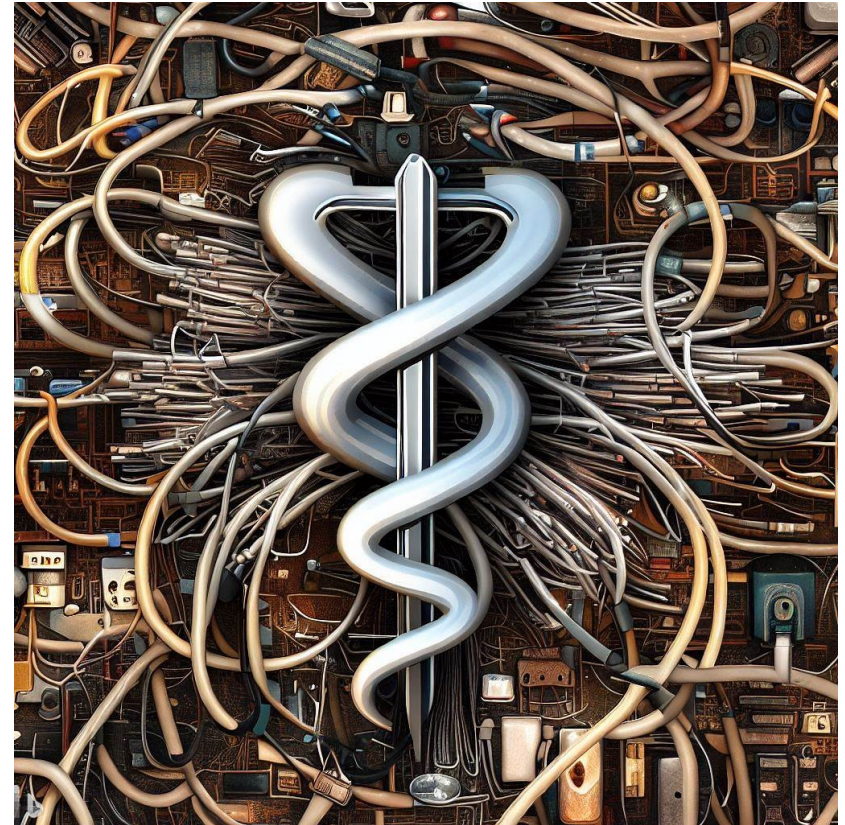
Take any MAC and glue SPE on it!

- Every Ethernet controller can support 10/100/1000BaseT1* PHYs.
- Every switch controller with external PHYs, can support *BaseT1* PHYs.



Network trends in Medical Devices

- Transition from proprietary to standardized Ethernet protocols, PTP.
- Explore Single Pair Ethernet (SPE) to reduce cable needs.



Network trends in Industrial Devices

- Shift from Fieldbus and CAN to 10BaseT1L Ethernet.
- Utilize existing cable infrastructure with new protocols.
- Use PoDL to reduce even more cables.



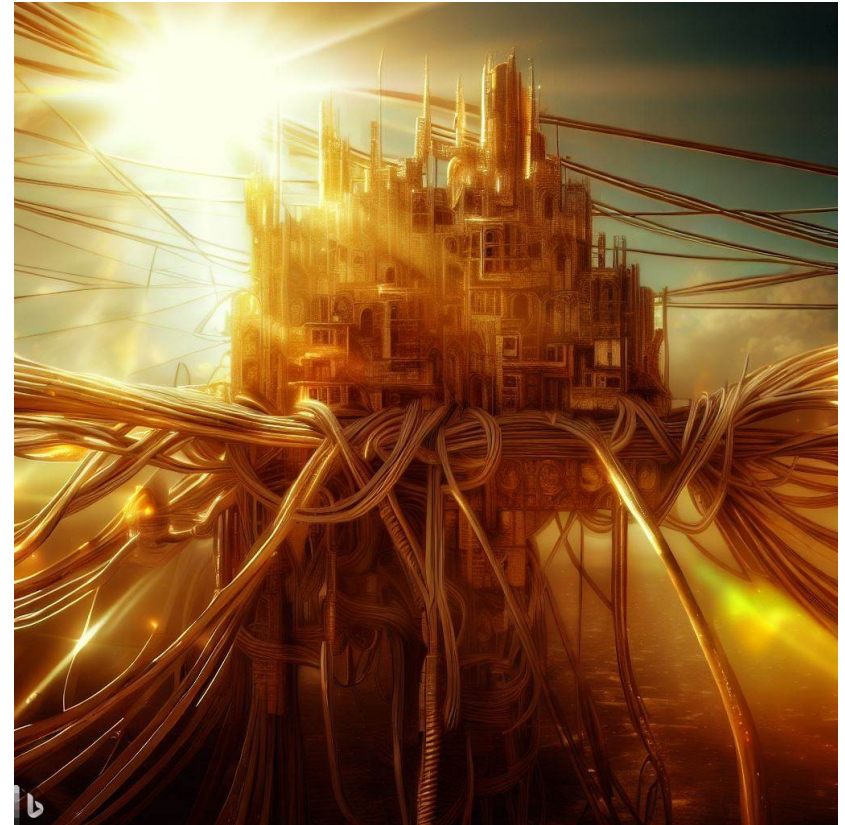
Network trends in Agricultural Devices

- Provide internet access, ensure security, handle updates, and offer long-term support.
- Adoption of Ethernet technologies (10BaseT1S, 100BaseT1, 1000BaseT1) for unified development.



Common Network trends

- Optimizing development resources and reducing the variety of different protocols by adopting Ethernet-based technologies and SPE.
- This enables a more efficient, secure, and maintainable network environment across different sectors



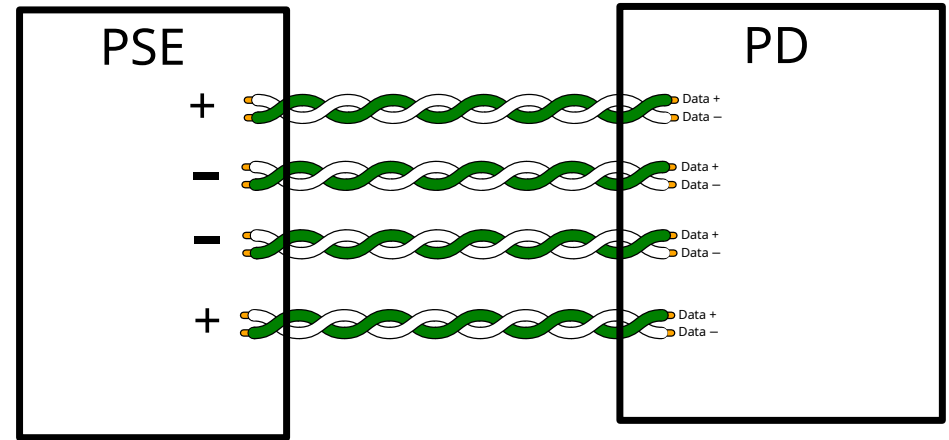
Making Green Tech Mainstream in Linux

- EEE support. Mainline but need a lot of fixes
 - <https://lore.kernel.org/all/20230331005518.2134652-1-andrew@lun.n.ch/>
 - Your help is needed, see last part of presentation.
- SPE: 1000BaseT1, 100BaseT1, 10BaseT1L, 10BaseT1S
 - With kernel v6.4-rc1 should be every thing mainline
 - Ethtool v6.3 has some extras for 10BaseT1S
- PoDL: Power over Data Line
 - With kernel v6.4-rc1 should be every thing mainline
 - With ethtool v6.3 supported



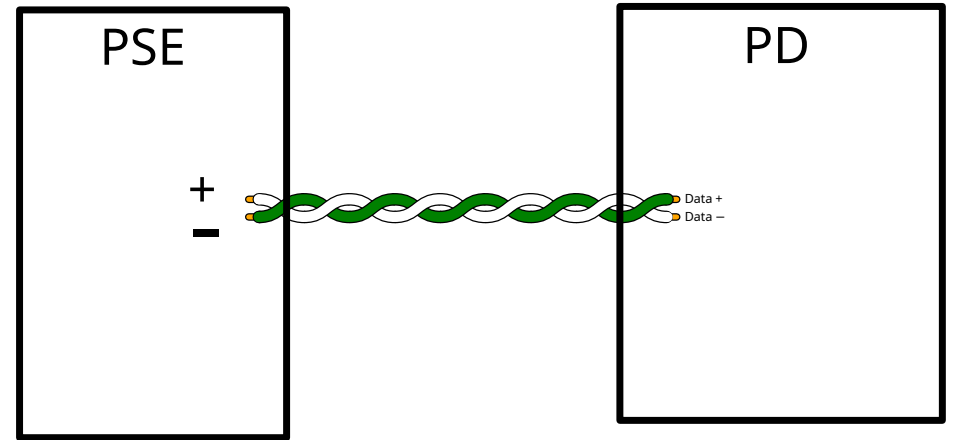
What is PoE4?

For power delivery are used
all 4 pairs



What is PoDL?

- Power over Data Line
- Same as PoE, except power is delivered over one twisted pair
- PoDL needs different HW implementation compared to PoE* variants



Detection

- PD is hot-plugable
- PSE provides a constant current on the line and measures the voltage drop if PD is connected
- Optional



Classification

- Classify, how much Voltage and Current should and can PSE provide to the PD.
- PoE and PoDL have different way for classification.
- PoE uses a constant voltage from the PSE and a current sink on the PD to signal the power class. Cable type may affect classification results.
- PoDL is using SCCP – simple, digital, one-wire communication serial classification protocol.
- Optional functionality



Current kernel implementation

- Provides per network port ethtool interface
- Currently we can:
 - control admin state of PSE per port independent of link admin state
 - Read admin state and status of PSE



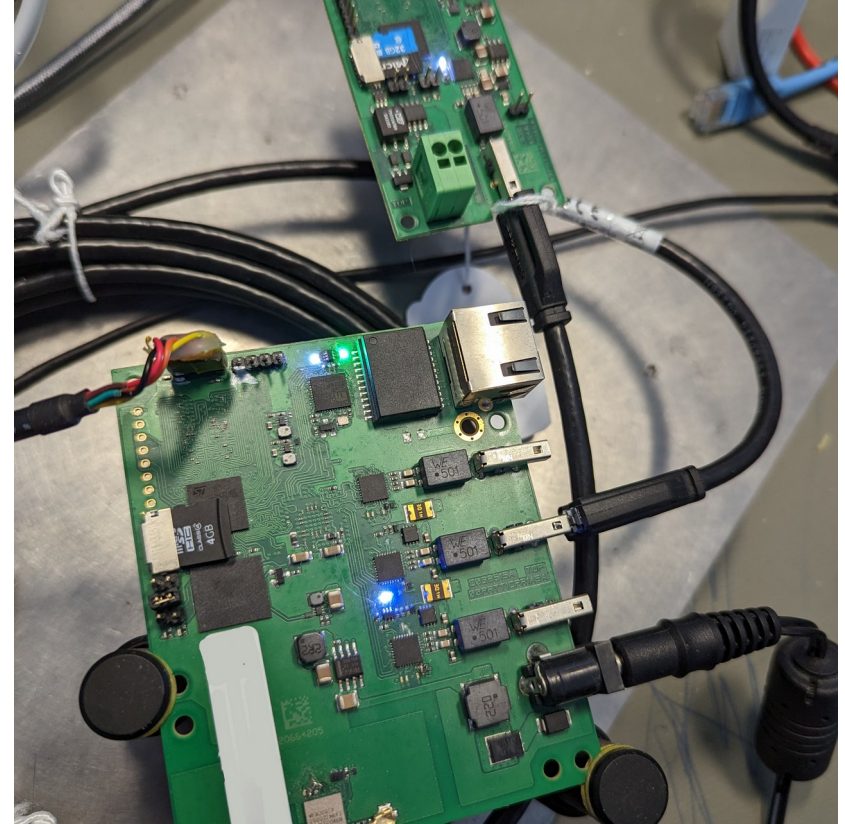
ethtool implementation

- `$ ip l`
- ...
- `5: t1l1@eth0: <BROADCAST,MULTICAST> ..`
- ...
- `$ ethtool --show-pse t1l1`
- PSE attributs for t1l1:
- PoDL PSE Admin State: disabled
- PoDL PSE Power Detection Status: disabled
-
- `$ ethtool --set-pse t1l1 podl-pse-admin-control enable`
- `$ ethtool --show-pse t1l1`
- PSE attributs for t1l1:
- PoDL PSE Admin State: enabled
- PoDL PSE Power Detection Status: delivering power



10BaseT1L switch with PoDL

- Real HW using SPE and PoDL
- Size difference between RJ45 and SPE connector.



Reducing power consumption - EEE

- Energy Efficient Ethernet
- On some systems, EEE saves 0.2W per port
- One Watt Initiative – reduce standby power under one watt.



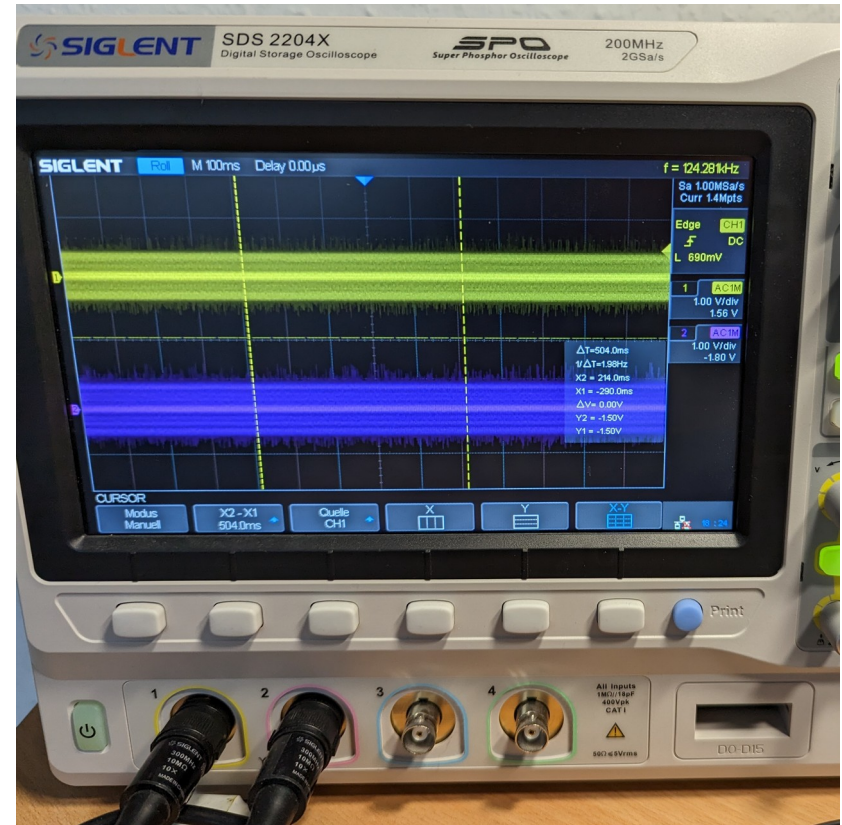
Trouble shooting EEE

- Current state of EEE support in Linux kernel v6.3 is different. Some drivers do it properly.
- Drivers or even HW may provide not enough or not proper information.
- Use oscilloscope!
- This talk is to inspire more kernel hackers to explore this functionality



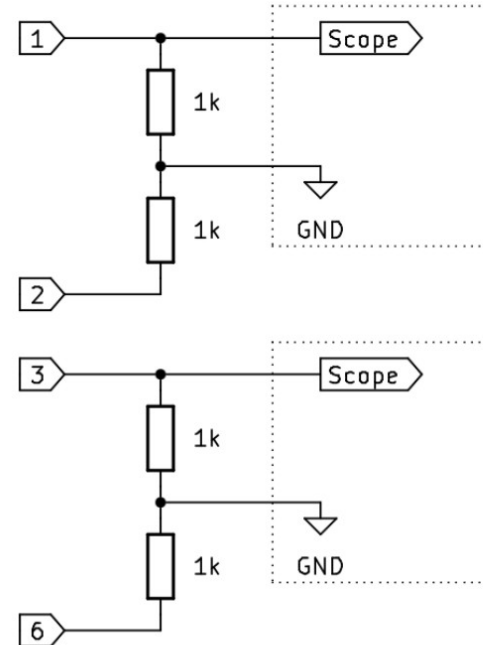
Oscilloscope

- No upper budget limit
- Let's reduce budget to get more hackers on board :)
- No 1000BaseT or 100BaseT decoder support is need
- It is enough to presence of the signal, not exact form of it.
- 2x channels is enough



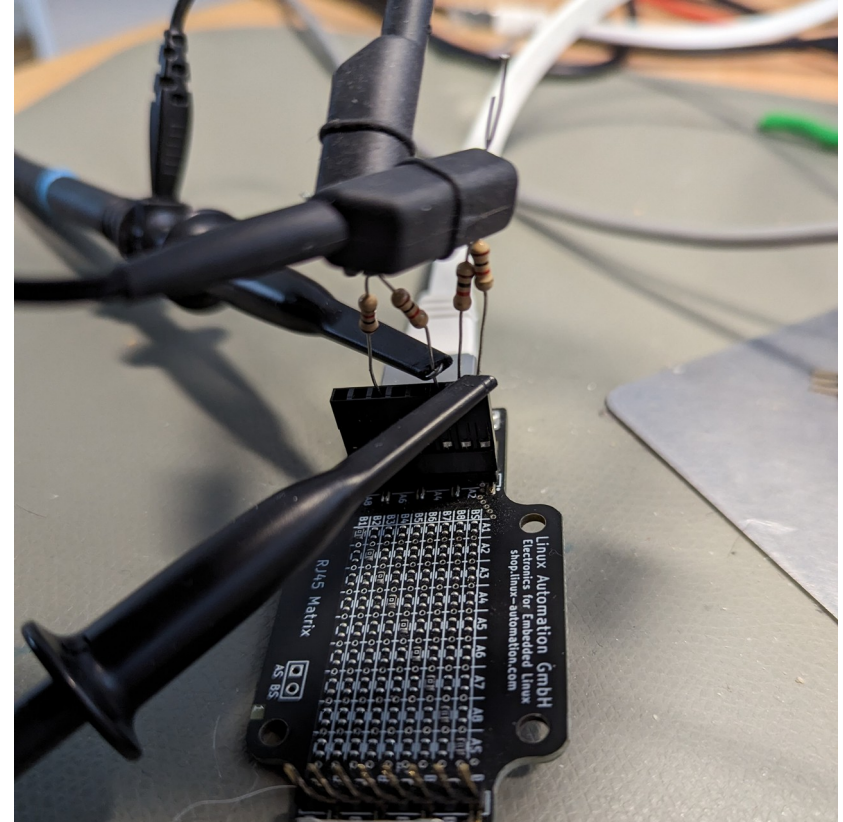
Probes

- Normally - differential probe is needed
- But we are doing low budget setup, so let's use bunch of 1kOhm
- If you know your HW setup it should be less risky to do so.
- Be careful to avoid HW damage!!
- Make sure no PoE or PoDL is in use!!!



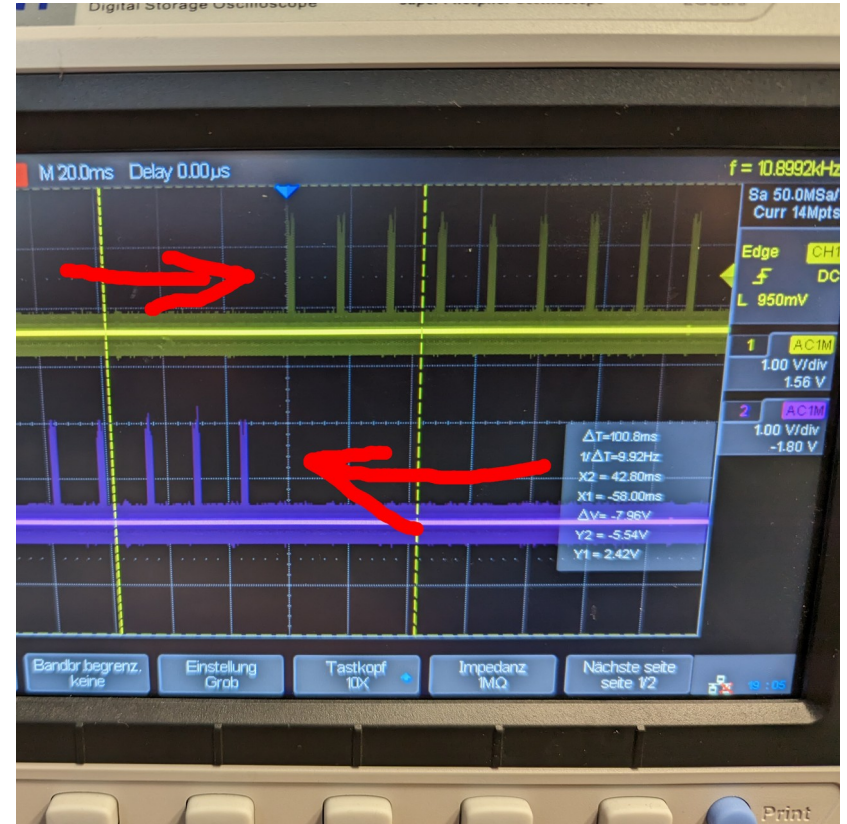
Probes – reducing noise

- Without differential probes and too long wires there will be too much noise
- Optimizing it a bit will make this setup more usable.



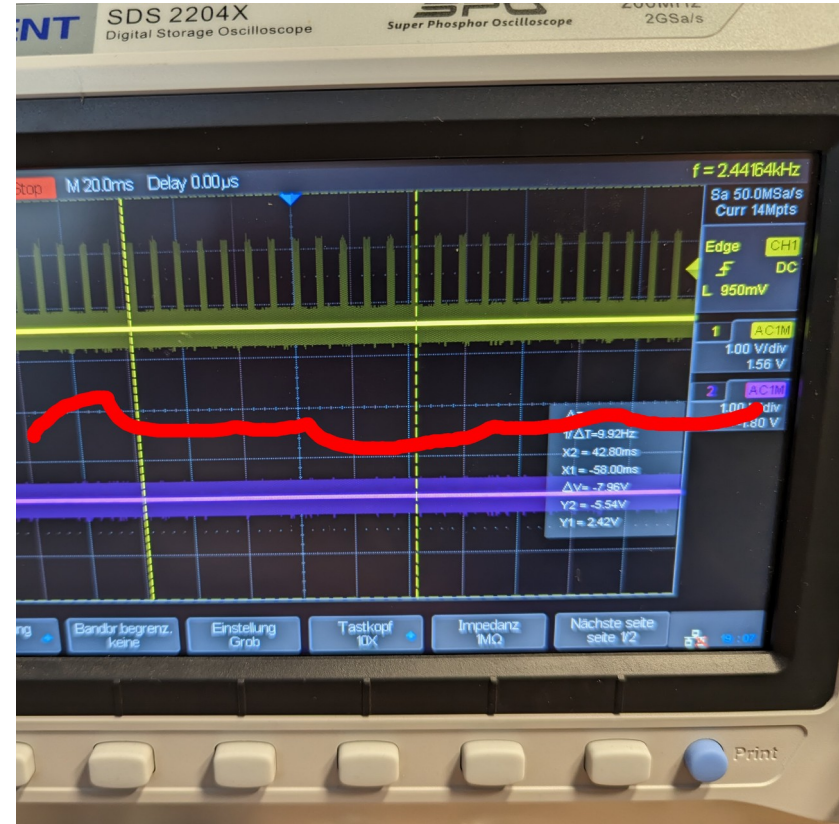
Get MDI-X under control

- First attach only one device
- `ip l s dev eth0 up`
- If pulse on both channels auto MDI-X is active
- Disable it to make things predictable



ethtool -s eth0 advertise 0x008 mdix on

- Some pre-configuration
- “advertise 0x008” – advertise only 100BaseT/Full. It is easier to debug with low budget setup
- “mdix on” – force MDI-X configuration. Not auto MDI-X. Link partner should stay Auto or depending on cable “mdix off”
- If mdix off/on is not working. Send patches :)



ethtool --show-eee eth0

EEE settings for eth0:

EEE status: enabled - active

Tx LPI: 500040 (us)

Supported EEE link modes: 100baseT/Full

1000baseT/Full

Advertised EEE link modes: 100baseT/Full

1000baseT/Full

Link partner advertised EEE link modes: 100baseT/Full

1000baseT/Full

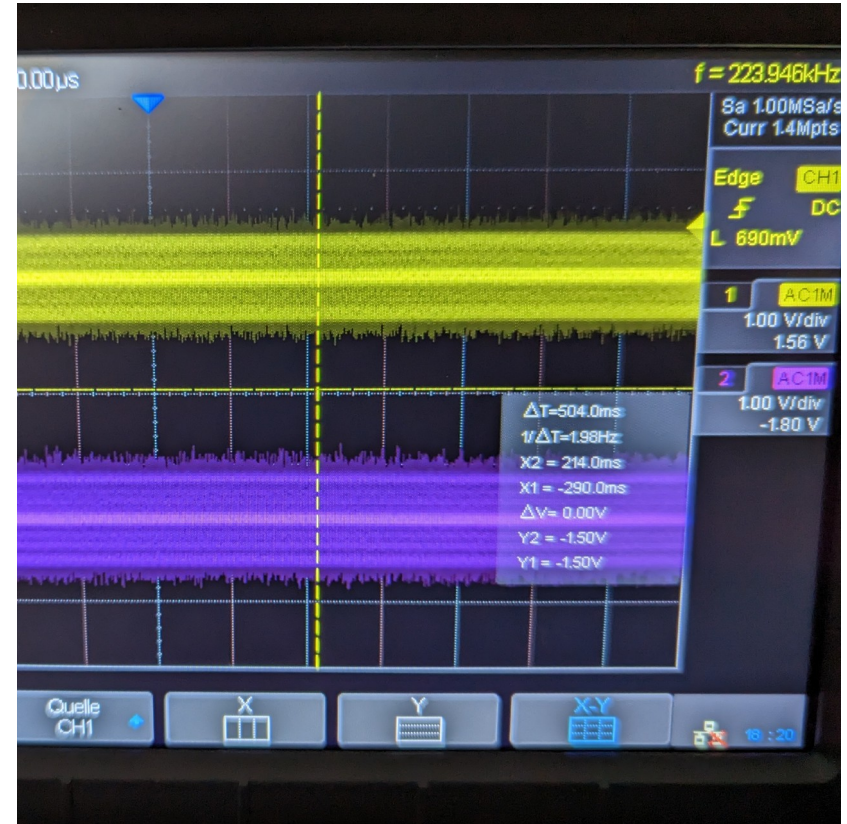
ethtool --set-eee eth0 eee on

- If “EEE status: enabled – active”. We should get some how similar picture
- There are no active transfers on the link



ethtool --set-eee eth0 eee off

- If “EEE status: disabled”
- Or “EEE status: enabled - inactive”

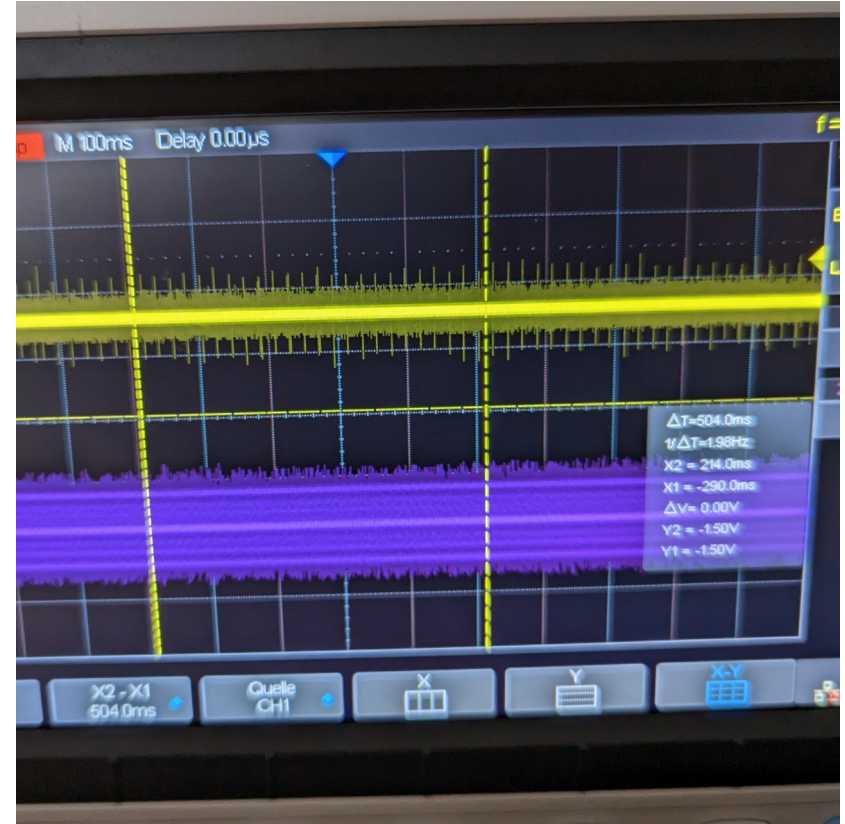


ethtool --set-eee eth0 eee on <> off



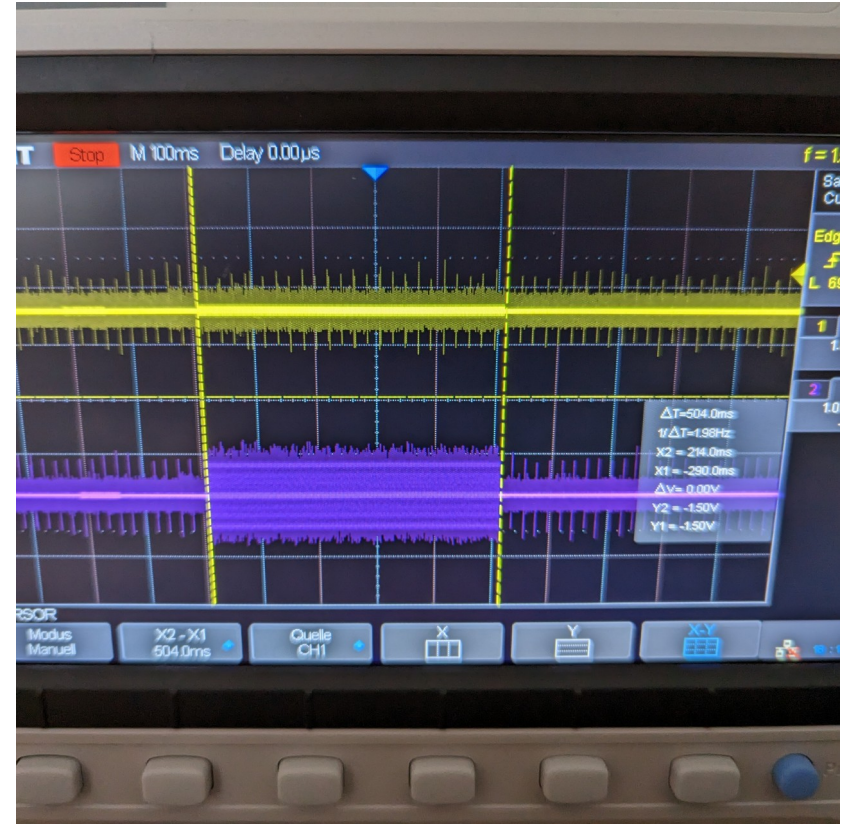
ethtool --set-eee eth0 tx-lpi off

- LPI – Low Power Idle
- It is possible to partially disable EEE
- tx-lpi off – disable TX LPI on local side
- By default - tx-lpi on



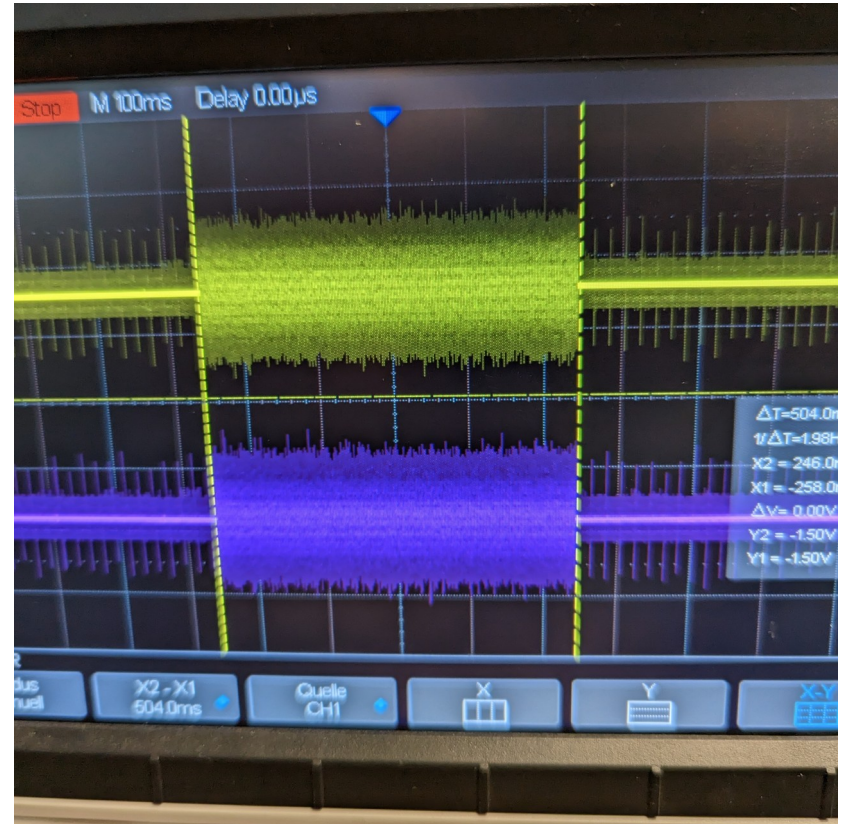
ethtool --set-eee eth0 tx-timer 500000

- Tx-timer – how long we should not enter LPI after transmission
- Send some packet to test this state. For example:
mausezahn eth0 -c 1 -a
rand -p 64



ethtool -s eth0 advertise 0x020

- Compare if things look similar with 1000BaseT
- advertise 0x020 – advertise support only for 1000BaseT/Full
- Note: with 1Gbit same ping will appear on both channels



Thank you!

Questions?

